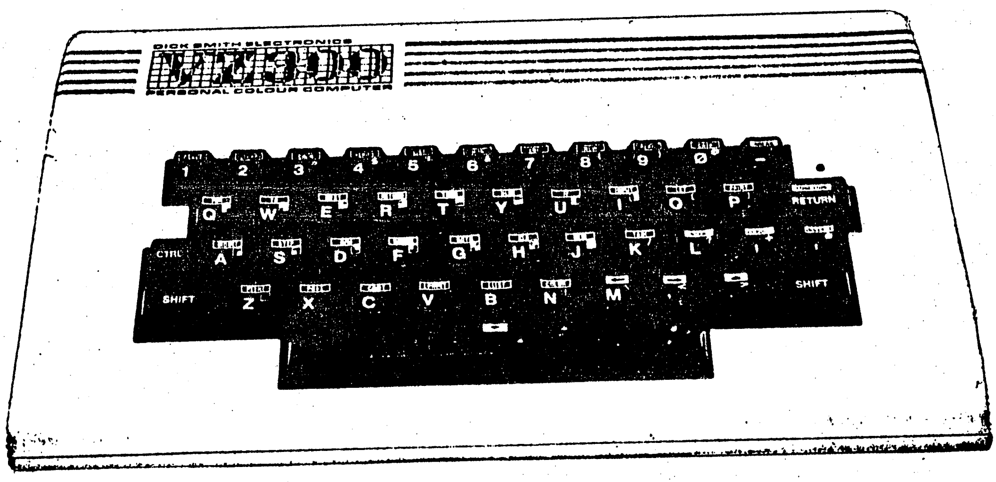
ISSUE NUMBER 7.

HUNTER VALLEY

VZ JOURNAL



DECEMBER 1986

PRESIDENT . . . . . PETER ELLIS

3 GOW STREET
HAMILTON NORTH
N.S.W.   2303
(049) 69 5697

SECRETARY . . . . . ROSS WOODS

83 LAMBTON PARADE
SWANSEA HEADS
N.S.W.   2281
(049) 71 2843

VICE PRESIDENT . . MARK O'BRIEN

46 FERN STREET
ISLINGTON
N.S.W.   2296
(049) 61 5490

TREASURER . . . . LEIGH ROGERS

40 FLEET STREET
NEW LAMBTON
N.S.W   2305
(049) 57 5738

EDITOR . . . .

POSITION VACANT

MAIL ALL SUBMISSIONS TO :-

HUNTER VALLEY VZ USERS' GROUP
C/- P.O. BOX 161 JESMOND
N.S.W   2299

PRESIDENT'S REPORT - - -

Hello there, welcome to the new look magazine. There have been some interesting developments during the last month and we hope these will teach us a few lessons and allow a whole new outlook for us as a group and a magazine. During the last few months this magazine has tended to lean more towards the advanced side of computers and while this is great if you're a hacker not everyone will benefit from a one sided look at things. Therefore future issues will attempt to be more balanced with a bit to please most people.

There should be a new name on the cover this month. We concluded that the old name was a bit of a mouthful and decided a shorter name was about due. In this issue you may notice the format is different to the others, the reason for this is that our previous editor is no longer with us, But we will struggle on regardless and a continuation of past articles should appear in future issues.

In this issue we have put a questionare for you to fill out. This is not compulsory but we would like you to fill out what you think is suitable to you and send it back to us. This will allow us to find out what range of membership we have and how we can best cater for your needs. There is no need to rush your answer back to us, take your time and think about it over Christmas and send it back in the New Year. In the meantime, do you have any articles, programs, technical projects that you have written and would like to share with us? If you would like to see your work in print then send it in to us and we'll be glad to look it over and print it if it's suitable. All articles will have the writers name attached to them, so you may be sure of full recognition for your work.

For the local members who attend our monthly meetings, dont forget our Christmas party this month. This will be a bring a plate affair because the bank account is in need of a respirator. Well, thats enough for now so read on and see what you think. If the quality of the magazine is not our normal high standard please bear with us as this will improve as we get everything sorted out.

On behalf of The Commitee of the H.V.V.Z.U.G I would like to wish all members and their families a Merry Christmas and a very happy New Year.

Bye for now.
Peter Ellis
[President]

The commitee of the H.V.V.Z.U.G. wish to advise that Gavin Wiliamson no longer represents above committe.

SECRETARIES REPORT

I haven't had much to say in previous issues, mainly because of my involvment in other areas than COMPUTERS. I will endeavour to correct this in future issues.

At the NOVEMBER MEETING of H.V.V.Z.U.G. Several Members expressed the need for changes to the MEETING'S and NEWSLETTER CONTENTS. To better gauge member's ideas it would be appreciated if a QUESTIONAIRE with this issue could be returned to me.

ONE of the views raised at the last meeting was for more articles of a simpler nature, we will try to rectify this without decreasing present types of articles. Also we will try to have LLISTings on Disk and Tape of the same month's issue as the meeting. As we also have subscribers we could possibly introduce some form of service to them as well.

IT would be an advantage to the author and editor if we got feedback from members about any articles appearing within these pages. Your comments, suggestions and constructive criticism would be most welcome.

I HOPE YOU CAN FIND VZ'S FUN AND OF SOME FUTURES USE.

ROSS WOODS.

SPECIAL NOTES :- By Joe Leon

First as you may already know our previous Editor Gavin Williamson is no longer with us. The club owes him a debt of gratitude for helping get this publication of the ground. A job well done gavin. AS acting Editor (no ham jokes please) appointed by the committee and with their help and support we'll endeavour to keep up the standard set and if possible improve on it. To that end we ask our Members, Suscribers and especially our Contributors both old and new for their continued support.

My thanks to all who helped put this publication together at short notice. Some regular articles had to be postponed due to lack of material and some although on way did not arrive in time for publication.

APOLOGY to DAVE MITCHELL :- In November's issue of the Technical Report on Soft Start Interrupt Joe Leon was wrongly credited with the schematic which was D.Mitchell's. As mentioned in the Editorial Joe Leon only prepared article for publication and as he did not have opportunity to preview before publication the error could not be corrected. My apologies Dave. Joe Leon

Exept for the LLISTINGS this Journal was produced using D.Smiths W.Processor which is the prefered medium. We would be most happy if intending contributors could oblige. If not possible then material should be presented on A4 size paper.

NOTE :- If Article or LLISTing is submitted on Tape please include a printout as well. A games review arrived on tape and although three Datassettes were tried they all failed to load review.

HUMOROUS ? . . To ERR is HUMAN, to really foul things up you need aCOMPUTER.

PRINT/LPRINT SWITCH BY R.QUINN

There is a single memory cell in the communications region of reserved RAM, with address 30876, which is used by your VZ as a PRINT/LPRINT switch. Your VZ is in PRINT (send data to the screen) mode when this cell contains a '0', and in LPRINT (send data to PRINTer) mode when it contains a '1' (or any non zero number between 1 and 127). '0' is the default byte for this cell, to which it will revert at the end of execution of a PRINT or LPRINT statement.

POKE30876,1:PRINT"TEST":PRINT"AGAIN?" will LPRINT the data of the first PRINT statement to your printer, even though you used the PRINT command. PRINT goes along with whatever mode is dictated by the POKE30876. The second PRINT statement sends its data to the screen. POKE 30876,0:LPRINT PEEK(30876) however, will not PRINT to your screen. LPRINT forces the switch to LPRINT mode. And of what use is all this? Well I have found two uses so far. Two VZ DOS (disk operating system) commands--DIR and STATUS--involve PRINTing certain disk data to the screen. Using the PRINT/LPRINT switch, we can use these commands to direct the disk data to your printer instead. With a disk in your drive and your printer switched on, enter the following sequence of direct commands: POKE30876,1:DIR:STATUS <RETURN>. So long as the two disk commands follow one another in this way, they are treated as a single LPRINT command.

The second use is available to all VZ users (not just those fortunate to have disk drive). I'll explain it with an example. I adapted a TRS-80 DISASSEMBLER program to run on the VZ. The program involves a loop with hundreds of conditional PRINT statements of which only a few are executed in each sweep through the loop. When I wanted an LPRINT version of the program so that I could record hardcopy of disassemblies, the only solution I could think of at the time was to make up a separate copy of the program in which all those conditional PRINT statements were changed to LPRINT statements. Which meant choosing which version to load depending on whether you wanted to PRINT or LPRINT. An awkward solution at best. I eventually devised a PRINT/LPRINT switch within the one version of the program using a routine similar to the COPY command. But a third solution arises, making use of the POKE 30876 switch. An INKEY$ switch would be used to choose between PRINT/LPRINT mode.

Line no. A$=INKEY$:IFA$="V"THENW=1ELSEIFA$="P"THENW=0. Then you only need place the following statement at a few strategic locations in the program--at the entrances to various blocks of conditional PRINT statements: POKE30876,W and on each sweep through the loop the program will PRINT or LPRINT the few PRINT statements which meet the current conditions according to the value of W. If you adapt this technique to one of your programs, remember that the POKE 30876,1 will remain in LPRINT mode only for the execution of one PRINT statement. With a little refining you can get a program to repeat each sweep through a loop so that it PRINTs the data to the screen on the first sweep, and LPRINTs the data to your printer on the second (repeat) sweep.

ED. NOTE :- For an example using above technique see line 340 in the DISK MENU MAKER in this issue.

This article was written with the "month-old" beginner in mind. We all know how you feel: you've just bought your new toy, you've played a few games, and now you'd like to know what else it can do.

Well, here's a short, simple programme for you to play with. Every step has been explained and even if you do know a bit more about programming, you might find it amusing anyway.

```
10 CLS
20 PRINT@266,"EAT AT JOE'S"
30 FOR A=0 TO 500
40 NEXT A
50 CLS
60 FOR B=0 TO 500
70 NEXT B
80 GOTO 20
```

This programme merely flashes "EAT AT JOE'S" in the middle of the screen; just like the old Donald Duck cartoons you saw as a kid. Neat eh! (I thought it was cute).

You may well be familiar with some of the commands, but just in case, they are explained.

```
10 CLS
```

This will clear the screen in preparation for the data to come. It is advisable to always use this command first with any programme you write - it's like getting a clean piece of paper.

```
20 PRINT@266,"EAT AT JOE'S"
```

This one is more complex. To understand this, you will have to imagine that your screen is like a grid and has been divided into 512 little boxes. Each box holds one letter, and they have been numbered across the screen. By inserting this statement, we have told the computer to, "Print at box No.266, this message, "EAT AT JOE'S" Take note of the commas and the like; you don't have to know what they are for, just don't leave them out (I call it Computer Gramma).

With a bit of mucking around, you will discover that "PRINT@ 266,"EAT AT JOE'S" prints the statement in the middle of the screen, and "PRINT@ 20,"EAT AT JOE'S" will place the statement further up the screen. In fact, you can place it anywhere on the screen between the numbers 0 and 511, (0 is a number too).

```
30 FOR A=0 TO 500
40 NEXT A
```

This is known as a "FOR, NEXT" loop. Don't be scared off by the technicalities of the name. All you are doing is telling the computer to count to itself, thus only waiting long enough so you can see what it has just displayed. You tell it to count by saying that A=0 TO 500, what is the NEXT A? 2, Right? NEXT A? 3... Until 500 is reached. You make it count a smaller number, it will wait less time and thus flash faster.

FOR BEGINNERS — CONT. - -

```
50 CLS      - Now clear the screen again...
60 FOR B=0 TO 500
70 NEXT B - and wait again.
```

Notice that "B" is used in this loop instead of "A" as the first "A" may be confused with the second "A". For a programme with more than one loop, a different character (or variable) must be used for each loop.
80 GOTO 20 - Just that... Go to line 20. At 20, it again prints "EAT AT JOE'S".

If You look at the programme as a whole you'll find that LINE 10 is executed only once while the rest is in a continous loop, because LINE 80 tells it to go back to LINE 20 and do it all over again.

I have found that the only way to master programming is to play with small programmes such as this. You can expand it by inserting a different statement as in line 20, and then another loop as in lines 30 and 40 (use "C" this time). This will make the screen alternate or "toggle" between the two statements.

Have fun playing and I'll see you all in the next issue.

PS. I'd like an idea as to what variety of programmes you mob out there would like to see, as this is my first contribution to the magazine.

PRINTER PLOTTER - - -

The P/Plotter section will be a short one this month. Mainly a warning and some tips.

First the warning. If you are getting your paper rolls from D.Smiths be aware that the end of the of the paper roll is glued to the plastic core which is in two sections. Attempts at continued line feeds by the Plotter cracked the motor gear wheel when the end of the roll was reached. There were no problems with the larger paper rolls previously available from D.SMITH'S. I rang the service department, but they did not carry any spare parts exept for a complete Plotter module at $120.00. Spares suggested it would be cheaper to buy a brand new Plotter at $99.00. I eventually managed to pick a motor gear wheel from TANDY'S for $1.00.

I now buy my paper rolls from TANDY'S which are much larger in diameter and no glue is used so damage to your plotter is practically impossible.

PLOTTER PEN TIP :-
This one came from Dave Boyce. It is possible to top up your pens with stamp pad ink. I tried it and it works well while others did'nt have much success. With a pair of pliers pull out the nib from the pen body and use about 3 to 5 drops of ink. One thing to remeber is to make sure that when the pen is reassembled it is the same lenght as before. Thanks for the tip Dave.

Joe Leon

MENU MAKER NOTES :- by Joe Leon

Last months version of MENU MAKER had a couple of errors which have been corrected in this months version. Also a DIR LPRINT routine has been incorporated as well as visual and audible indication of MENU ERAsing, SAVING and LOADing. I thought it wortwhile to compare the differences in both also as they contain some usefull routines and programming techniques which some VZ Users may not be familiar with.

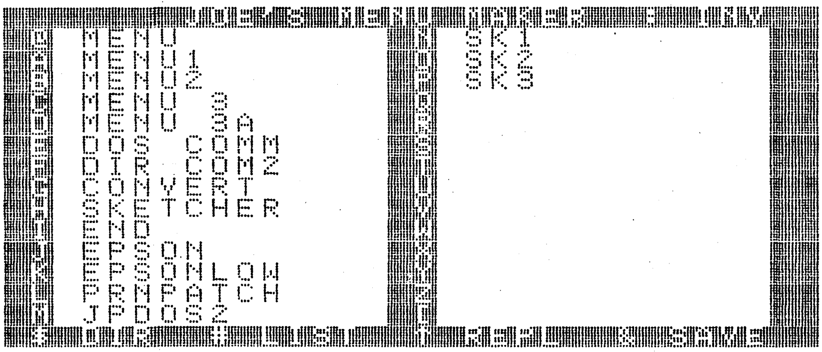
I've been asked why the choice of '@' and '[' in the MENU. The answer is very simple. '@' precedes 'A' while '[' follows 'Z'. It makes the coding simpler in LINE 200, also a maximum of 28 file names are catered for which is most desirable. If you divide 112 X 28 you should get 4. 112 is the maximum no. of files the VZ Dos will handle. It then follows that PAGING can be incorporated requiring just a little extra bit of coding which will be given next issue.

And now to the errors in last months version.
Line 40:- It was underDIMensioned, The '*' in LINE 400 wasn't accounted for. Just change the two 28's in LINE 40 to 29. The only time a problem arose if you had 28 files in the menu and you pressed '[' to load 28'th file a crash would result.
LINE 60 had some missing code. Retype as per this issue.
If we use the screen dump as an example it will be easier to show how a crash could occur. To load the last file in the MENU, SK3 you would press 'P' to load. no problems there, but if you pressed 'Q' then a crash would occur. N=N-1 in LINE 60 was left out. The opportunity was taken to tidy up the coding in LINE 60 and the rest of the program as well.

MEMORY SAVING :-
A total of 60 bytes can be saved in LINES 250 to 280. Delete POKE30777,35: from above LINES, also :SAVE"MENU":RUN"MENU" from LINE 260 as it's already in line 270. Finally add :POKE30777,35 to LINE 130 after GOSUB310. Only one POKE is needed for the job. By using IF..THEN..ELSE on LINES 160 to 180, the three LINES can be condensed to one. It is harder to read, but saves on memory as well. See LINE 160 in this months version as an example.

And now to this months version. All the changes are from LINES 260 to end. Do not type in the DATA in LINES 400 to 420. They are there simply as an example. I find it very useful to have MENU,T as the first file in line 400 as it allows me to hop from one DISK MENU to another. There are three major changes.
1) LINES 260 to 275 give audible and visual indication of MENU ERAsing, SAVING and LOADing. I thought it desirable to know what was going on.
2) DIR LPRINT option has been included. It makes it a lot easier to enter filenames in DATA statements from a printout than memory. The POKE in line 340 makes it possible. Read PRINT/LPRINT SWITCH by ROBERT QUINN in this issue for an insight on how it works and ideas on how to use it in your own programs.
3) PRINTER STATUS CHECK :- How many times have you issued a print command and had the program crash or hang up simply because you forgot to connect it and or turn it on. I've been caught many times so now I include a PRINTER STATUS check routine in my programs. To check for PRINTER STATUS just type in immediate mode :- PRINTINP(0) ,the answer you will get may be different to the two in LINE 330. You will get different numbers with printer switched on and off, and also with different VZ'S. Ihave two VZ 200'S, one has 2 X 8K ROMS while the other has one 16K ROM and I get different nnmbers with each. Just use the number you get with your VZ. Only one INP test is required if you only have one VZ.

MENU MAKER BY PAUL AND JOE LEON




```
10 GOTO40
20  RUN"         "
30 '"
40 DIMP$(29),F$(29)
50 COLOR,0:POKE30744,96:POKE30862,80:POKE30863,52:POKE30777,1
60 N=N+1:READP$(N),F$(N):IFF$(N)<>"*"THEN60ELSEN=N-1
70 CLS:PRINT"";
80 FOR I=1 TO 14
90 PRINT@(32*I),""CHR$(191+I)" "P$(I);
100 PRINTTAB(15)""CHR$(205+I)" "P$(I+14)TAB(30)"";
110 NEXT
120 PRINT"";:POKE29183,96
130 POKE30777,35:GOSUB370
140 X$=INKEY$:X$=INKEY$:IFX$=""THEN140ELSEX=USR(X)
150 IFX$=":",POKE30744,ABS(PEEK(30744)=0)*96:SOUND20,1
160 IFX$="^"THEN260ELSEIFX$="&"THEN270ELSEIFX$="$"THEN280
190 IFX$="#"THENCLS:LIST400-
200 IFX$<"@"ORX$>CHR$(N+63)THEN140
210 P=ASC(X$)-63:CLS:PRINT@480,"  "F$(P)": "P$(P);
220 GOSUB370
230 IFF$(P)="B"THENPOKE31477,66ELSEPOKE31477,32
240 FORB=1TO8:POKE31479+B,ASC(MID$(P$(P)+"        ",B,1)):NEXT
250 GOTO20
260 CLS:PRINT@480," T: MENU";:SOUND30,1;25,2:ERA"MENU"
270 CLS:PRINT@480," T: MENU";:SOUND30,1;25,2:SAVE"MENU"
275 CLS:PRINT@480," T: MENU";:SOUND30,1;25,2:RUN"MENU"
280 CLS:DIR:STATUS:PRINT@18,"";:GOSUB370
300 R$=INKEY$:R$=INKEY$:IFINKEY$=""THEN300ELSEX=USR(X)
310 IFR$="N"THENRUNELSEIFR$="Y"THEN330
320 GOTO300
330 IFINP(0)=254ORINP(0)=246THEN340ELSE350
340 POKE30876,1:DIR:STATUS:RUN
350 PRINT@485,"";
360 GOSUB370:SOUND0,9;0,5:RUN
370 SOUND20,1;25,1:RETURN
380 '""
390 '""
400 DATA MENU,T,MENU1,T,MENU2,T,MENU 3,T,MENU 3A,T,DOS COMM,T
410 DATA DIR COM2,T,CONVERT,T,SKETCHER,T,END,T,EPSON,B
420 DATA EPSONLOW,B,PRNPATCH,B,JPDOS2,B,SK1,T,SK2,T,SK3,T
500 DATA,*
```

UNDERSTANDING YOUR VZ . . PT.5

By R.QUINN

POS will give you the current location of the cursor in whatever pair of display lines it has moved to. This is useful for keeping track of the cursor while a program is running since the cursor is invisible under program RUN. However, POS is not always reliable, and since it returns values in the range 0 to 63 only, it doesn't tell you what pair of display lines the cursor is in. Fortunately there is a pointer in the communications region which keeps track of the cursor in terms of its absolute address in text mode video memory (addresses 28672 to 29183). The pointer to cursor position occupies addresses 30752 and 30753, and can be accessed thus: X=PEEK(30752)+PEEK(30753)*256

You can use the value of X (or whatever numeric variable you choose to use) in your programs to control the display with absolute certainty as to where you're at. Of course, if you PRINT X then the cursor will no longer be at the address X, will it? X = X - 28672 will then return the cursor position in PRINT@ format where screen addresses have a range of 0 to 511.

STRING$ allows you to create and display a string of umpteen identical characters. The STRING$ format is STRINS$(b,c) where B is a number in the range 0 to 255, or a numeric variable, which determines the number of characters in the string. C can be a number or a numeric variable with a range 0 to 255, or it can be a string or a string variable. As a number, C is interpreted as a VZ ASCII character code, converting to CHR$(C). Examples: X$=STRING$(32,95) PRINT STRINS(255,140) 'This displays a string of 255 identical graphic characters. You will need to CLEAR enough memory for string space to accomodate long strings like this. If C as a string or string variable has more than one character, STRING$ will use the first character of the string.

PRINTSTRING$(X,A$) STRING$ can also be used with PRINT@

You can experiment with the STRING$ statement with this one line program:

```
10 INPUTB:INPUTC:FORR=1TO15:PRINTLET(B,C):NEXT:GOTO10
```

Now enter these commands: CLEAR999:POKE31484,196:RUN

At, and marking, the end of a basic program in program memory are two zero bytes. The last line in a program points (with its start address of next line pointer) to the first of these zero bytes as the start of the next line. But this 'next' line is a fake. Only these first two bytes of this pseudo line exist and they point to address zero. So you will always find three zero bytes at the end of a program--the zero byte marking end of last line, then the two zero bytes starting the pseudo last line. The next byte is the beginning of the VARIABLE LIST TABLE and it is the address of this byte that is pointed to by the END OF PROGRAM pointer, which is really a START OF VARIABLE LIST TABLE pointer. The VARIABLE list table lists all the variables--numeric and string--that are used when a program runs. Simple numeric variables and pointers to simple string variables are entered into the simple variables table on a first come first served basis. Array variables are likewise dealt with in the array variables table, which always starts at the end of the simple variables table. The first byte of each entry is a code which identifies the variable type: '2' is integer; '3' is string; '4' is single precision; '8' is double precision. The next two bytes code for the variable name.

VARPTR is one of the most useful of the basic words that were left out of the VZ's vocabulary. It can be used to explore the way in which variables are located, organised and handled in memory.

PRINTVARPTR (variable) will display the address of the fourth byte of the entry for the variable in the variable list table. Only simple string variables will be considered further here.

Z = VARPTR(A$)   Z is the address where the length of A$ is stored.

X = PEEK(Z)   Then X is the length of A$.
The bytes with addresses Z + 1 and Z + 2 are then a pointer in the variable list table to the start address of the content of A$.

Y = PEEK(Z+1) + PEEK(Z+2) * 256   then Y is the start address of the content of A$.

With this information we can change the length of a string and the start address of a string's content.

```
POKE X,A  or POKEVARPTR(A$),A
POKEZ+1,B  or POKEVARPTR(A$)+1,B
POKEZ+2,C  or POKEVARPTR(A$)+2,C
```

New your VZ and enter line 20:
```
20 POKELET(X$)+1,0: POKELET(X$)+2,112
```

Now POKE31470,192:POKE31481,192 and enter line 10:
```
10 X$=LET(32,32)
```
Now POKE31472,196:CLS:RUN

Invoke inverse (CTRL :), then position cursor on top screen line and type in whatever characters you please. Bring the cursor down the screen and PRINTX$. What we have done is given X$ a length of 32 and pointed it to the top display line of video screen so that the content of X$ will be whatever characters occupy the first 32 cells of video memory. Inverse mode entry is required for alphabetical characters and a few other characters because of the difference between ASCII character codes and VZ PEEK/POKE character codes. I leave you to work it out and to explore the possibilities of pointing strings to any part of memory (ROM or RAM) with any length from 1 to 255 bytes. Design your own INPUT routines!

Some string pointers point to locations in program memory (A$ = "LITERAL STRINGS"); others point to locations in string space (B$ = "NON" + A$). Your VZ will reserve 50 cells at the top of user memory for string space. If you want to reserve more (or less) string space memory then you use the CLEAR n command where n is the amount you want.

Note :- Next issue will contain concluding chapter on UNDERSTANDING YOUR VZ by ROBERT QUINN.